

An Automated Weather Research and Forecasting (WRF)- Based Nowcasting System: Software Description

by Stephen F. Kirby, Brian P. Reen, and Robert E. Dumais Jr.

ARL-SR-0283

October 2013

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

White Sands Missile Range, NM 88005-5501

ARL-SR-0283**October 2013**

An Automated Weather Research and Forecasting (WRF)- Based Nowcasting System: Software Description

Stephen F. Kirby, Brian P. Reen, and Robert E. Dumais Jr.
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY) October 2013		2. REPORT TYPE Final		3. DATES COVERED (From - To) NA	
4. TITLE AND SUBTITLE An Automated Weather Research and Forecasting (WRF)-Based Nowcasting System: Software Description				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Stephen F. Kirby, Brian P. Reen, and Robert E. Dumais Jr.				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory Computational and Information Sciences Directorate Battlefield Environment Division (ATTN: RDRL-CIE-D) White Sands Missile Range, NM 88002-5501				8. PERFORMING ORGANIZATION REPORT NUMBER ARL-SR-0283	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT A Web service/Web interface software package has been engineered to address the need for an automated means to run the Weather Research and Forecasting (WRF) model with four-dimensional data assimilation. The user provides a model center point and selects a domain configuration and model start hour. Global Forecast System model data is then downloaded from the National Centers for Environmental Prediction to provide the initial conditions for a model run. In addition, surface and upper-air observations are acquired and are merged with the forecast fields for the first 6 h to provide the most realistic picture of the atmosphere possible. JavaServer Faces toolkit is used to generate the user interfaces in conjunction with HTML5. Glassfish Web server is employed for Web service implementation.					
15. SUBJECT TERMS WRF preprocessing system, WRF, web service, automated					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 24	19a. NAME OF RESPONSIBLE PERSON Stephen F. Kirby
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) (575) 678-7474

Contents

List of Figures	iv
Preface	v
1. Summary	1
2. Introduction	2
3. Background	3
4. Software Architecture	3
4.1 Oracle Glassfish Web Server	3
4.1.1 Starting Glassfish	3
4.1.2 Deploying the WRFEE.....	3
4.1.3 Glassfish Logs	3
4.2 WRFEE Software	4
4.2.1 The Layout	4
4.2.2 The Components.....	4
5. Obtaining the Software	12
6. References	13
List of Symbols, Abbreviations, and Acronyms	14
Distribution List	16

List of Figures

Figure 1. WRFEE program flow.....	4
Figure 2. Sample namelist.input file for 15.75-/5.25-/1.75-km nested domain configuration..	10

Preface

The U.S. Army Research Laboratory (ARL) Battlefield Environment Division (BED) has long sought an automated means to perform all of the tasks required to generate Weather Research and Forecasting (WRF) forecast grids, including gathering the large-scale initialization data and observation data, performing the WRF Preprocessing System (WPS) tasks, generating the WRF initial and boundary conditions, and lastly, running WRF using four-dimensional data assimilation (FDDA). To address this need, BED has engineered a Web service that automates the entire process, from large-scale initialization data acquisition, to execution of WPS via a Practical Extraction and Report Language (Perl) script, to observation retrieval and quality control, and finally, the generation of high-resolution WRF grids utilizing FDDA.

INTENTIONALLY LEFT BLANK.

1. Summary

A Web service (WS), called Weather Research and Forecasting (WRF) end-to-end (here after referred to as WRFEE), that is graphical user interface (GUI)-enabled has been developed to generate high-fidelity mesoscale model forecasts that automatically incorporate Global Forecast System (GFS) large-scale initialization data as well as Meteorological Assimilation Data Ingest System (MADIS) observations to perform four-dimensional data assimilation (FDDA) using the state-of-the-art WRF model.

WRFEE is built on the Oracle Glassfish Web server and employs the Oracle JavaServer Faces (JSF) toolkit for data display through the Web interface. JSF is composed of two parts: a Hypertext Markup Language (HTML) toolkit and a “core” toolkit to augment all of the native features of HTML.

WRFEE is based on the model-view-controller (MVC) paradigm whereby the model controls program flow (see figure 1), the view is the means of presenting the data to the user, and the controller determines what program path is taken next. In the case of WRFEE, the model consists of a Java bean, “WrfendtoendBean” that steps through the entire process leading up to a WRF FDDA run. The view component is mainly handled via JSF through calls made within the HTML pages. Control is maintained by the “action” part of JSF statements inside the HTML pages, which will direct program flow to the HTML page indicated.

The user will be directed through several HTML pages, first indicating the capability of WRFEE, then requesting user input including the center point to be used for the WRF FDDA forecast run, the particular 3-nest domain configuration desired, and lastly, the forecast start hour. Note that to preclude the issue of occasionally not being able to retrieve archived GFS and MADIS data, WRFEE is built strictly as a real-time system.

At this point, the WRFEE java bean takes over the processing. First an assessment of which GFS files to download is made based on the current system time, the model start time requested, and by allowing 2 h for the dissemination of any given GFS model run. Upon successful retrieval of the GFS data from the National Centers for Environmental Prediction (NCEP), WRF Preprocessing System (WPS) can be executed. WRFEE incorporates the “RUNWPSPLUS” code module (includes a Practical Extraction and Report Language [Perl] script and WPS); the Perl script directs the retrieval of MADIS observations and execution of an enhanced version of WPS (WRFEE currently uses version 3.4.1). Once the information regarding the type of observations required as well as model start/stop times and model domain criteria are incorporated into the “runwpsplus.config” file, the “runwpsplus.pl” Perl code may be run, which will fetch the MADIS observations requested, perform quality checks on the data and generate observation files for each domain, ready for ingest into WRF FDDA.

WRF FDDA is accomplished using a modified version of the WRF (3.4.1) model distributed by NCAR. WRF is compiled for distributed memory–parallel (DMPAR) (Message Passing Interface [MPI] is used and information is passed between nodes) rather than shared memory–parallel (SMPAR), (where open multiprocessor [OpenMP] is used) since it has been determined that WRF FDDA does not run correctly when executed in SMPAR mode. The modifications to WRF are discussed later in this report.

2. Introduction

To address the need for an automated software system that can generate high-resolution forecast grids that incorporate hourly observation data, a WS has been constructed. Upon successful startup of an Oracle Glassfish Web server instance, the user can deploy the “WRF end-to-end” Web archive (WAR) file. This allows the user to interact with the WS via the Web interface.

The user will step through a series of Extended Hypertext Markup Language (XHTML) files that will first inform the user of WRFEE capability, and then query the user through a Web interface for (1) a WRF domain center point, (2) a nested domain configuration, and (3) a model start time. At this point, program control transfers to the Java bean, “WrfendtoendBean”.

WrfendtoendBean ascertains which GFS files to fetch from NCEP, based on the current system time, the model start time requested, and with the built-in assumption that approximately 2 h should be allowed for NCEP to disseminate a given GFS model run.

Upon successful retrieval of the GFS large-scale initialization data, WPS (3.4.1) is run, orchestrated by the Perl script, “runwpsplus.pl”. “RUNWPSPLUS” is the name given to the entire package including the Perl script and accompanying WPS Fortran code from NCAR. The functionality of WPS will be described in detail in the “Wrfendtoend Java Bean” section. The configuration file required by the Perl script is automatically generated based on the user inputs regarding model domain geometry and location as well as the forecast start hour.

RUNWPSPLUS then proceeds to fetch MADIS observations matching the timeframe requested, performs quality control, and finally outputs them in time-order ready for ingest by WRF FDDA.

A Battlefield Environment Division (BED)-modified version of the NCAR-supplied WRF package (3.4.1) is used for WRFEE. One improvement incorporated into WRF is a fix to handle over-drying of the atmosphere when observation nudging is employed. All of the modifications are outlined later in this report.

3. Background

WRFEE is based on the MVC paradigm whereby the model handles data and the business logic. The view is what presents data to the user while the controller receives user requests and calls appropriate resources to carry them out (*I*).

In terms of WRFEE, the model is the Java bean, `WrfendtoendBean.java`, which orchestrates, based on user selections through the user interface XHTML pages, the entire process beginning with the determination of run location, domain configuration, and model start time, and finally, the execution of WRF FDDA.

The view is enabled by the use of JSF with its core and HTML libraries for building user interfaces to include entities such as buttons, tables, user fill-in boxes, etc.

The controller can collectively be considered all of the XHTML files, which when they receive data from the user interface XHTML page, redirects program control appropriately.

4. Software Architecture

4.1 Oracle Glassfish Web Server

To obtain the Glassfish Web Server, go to <http://www.oracle.com> and click the “Downloads” tab at the top. Look for the “Java” tab midway down the page and click on it. This will allow one to link to “Java EE & GlassFish Server”. Select the “Oracle GlassFish Server” option.

4.1.1 Starting Glassfish

Upon successful install of the Glassfish Web Server, a default domain, “domain1” may be started via this command: “(glassfish-install path)/bin/asadmin start-domain”. (To stop a domain, substitute stop-domain for start-domain.)

4.1.2 Deploying the WRFEE

In general, a deploy command will look like, “(install path)/bin/asadmin deploy war-name”, where war-name represents the WAR file containing your application.

4.1.3 Glassfish Logs

Once Glassfish Web server is started, the “server.log” provides information pertaining to server status. If the default domain, domain1, is being used, look at “(glassfish-install-path)/glassfish/domains/domain1/logs/server.log” for log information on the current domain being run.

4.2 WRFEE Software

4.2.1 The Layout

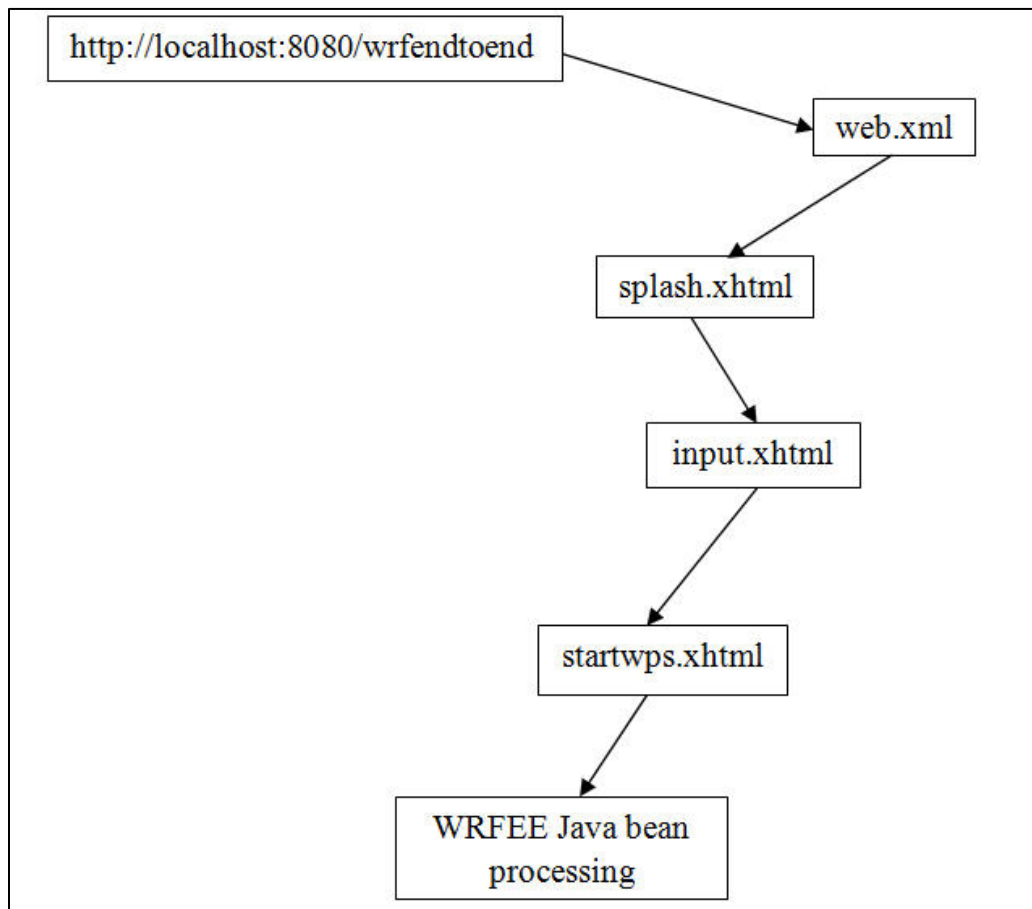


Figure 1. WRFEE program flow.

4.2.2 The Components

web.xml

An example of program control within WRFEE is exemplified by the “web.xml” file, wherein a “<welcome-file>” placeholder denotes which page (*.html, *.xhtml, *.jsp, etc.) to display first when the user types in the WRFEE URL. In the case where no welcome-file is designated, Glassfish server will search for a file named index.html. In the case of neither a welcome-file being designated nor an index.html file being present, Glassfish server will return a directory listing (2). For WRFEE, the <welcome-file> entity points to “splash.xhtml”. Within the web.xml file, the user can also designate how to map certain files with JSF (such as *.xhtml).

splash.xhtml

The “splash” page provides an overview of WRFEE capability to the user, indicating what inputs are required, what software modules are employed, and what output is generated.

This HTML page, as with all of the WRFEE HTML pages, employs JSF. JSF is incorporated into the HTML by two designations at the top of the page:

```
xmlns:h="http://java.sun.com/jsf/html"
xmlns:f="http://java.sun.com/jsf/core"
```

The first assignment above sets “h” to be a local variable representing the HTML portion of the JSF toolkit, while “f” is assigned to the “core” portion of JSF. More specifically, “h” represents “the tag library (that) contains JavaServer Faces component tags for all UIComponent + HTML RenderKit Renderer combinations defined in the JavaServer Faces Specification” while “f” represents “the core JavaServer Faces custom actions that are independent of any particular RenderKit” (3). Having been designated, the “h” and “f” local variables can then be invoked for the rendering of JSF toolkit entities. The action portion of the JSF block of code below signals to the WRFEE controller to transfer program focus to the input.xhtml page.

```
<h:form>
<h:commandButton value="#{labels.gotouserinputpage}"
action="input" />
</h:form>
```

Also note that the string displayed on the button is the string associated with the variable “gotouserinputpage” found in the (GLASSFISH_WEB_SERVICE_HOME)/WEB-INF/classes/bundle/labels.properties file.

input.xhtml

From this page, the user is first queried for a latitude and longitude value to serve as the coarse domain center value.

The user will then be requested to choose a nested domain configuration from among:

1. 15.75-km (113×113 points)/5.25-km (145×145 points)/1.75-km (289×289 points) nested domains—where 15.75, 5.25, and 1.75 refer to the horizontal grid resolution. This provides a $1764 \text{ km} \times 1764 \text{ km}$ outer area.
2. 9-km (175×175 points)/3-km (241×241 points)/1-km (127×127 points) nested domains, yielding a $1566 \text{ km} \times 1566 \text{ km}$ outer area.

Lastly, the user is required to select a model start time in hours Greenwich Mean Time (GMT), among 00, 06, 12, or 18 h. Note that WRFEE was purposefully designed as a “real-time” system to avoid the issue of unavailable data from either NCEP (large-scale initialization data) or MADIS (observations). Upon successful completion of these three options, program control is transferred to “startwps.xhtml”, via the call, “<h:commandButton...”>action=startwps.

startwps.xhtml

The “startwps” page displays the options chosen and informs the user of what processing will subsequently take place. When the user clicks the button, rendered from the JSF code, `<h:commandButton=“...“>action=wrfendtoendBean.generatewpsgrids`, program control transfers to the method, “generatewpsgrids()”, within the WRFEE Java bean, `WrfendtoendBean.java`.

Wrfendtoend Java Bean

The WRFEE bean represents the “model” portion of the MVC system, as it steps through each of the requisite processes, performing calculations, passing data blocks between procedures, all leading up to the WRF FDDA run.

In the java bean code, annotations such as “@WebMethod” are used. The purpose of this is to customize a method that is exposed as a WS operation (4).

The bean first cleans out any remnant observations in the MADIS repository directories as well as any files resident in the WPS and WRF “working” directories. The purpose of this step is to insure data files from a prior run of WRFEE do not contaminate the current forecast run.

Next a determination of the proper projection for WRF execution is made. The options are

1. Polar stereographic: 60deg < proj <= 90deg
2. Lambert conformal: 30deg < proj <= 60deg
3. Mercator: 0deg <= proj <= 30deg

An intermediate GFS base reference hour (i.e., what hour [Coordinated Universal Time (UTC)] the model is run for) is derived by accounting for these factors: (1) the model start time, supplied to WRFEE via the Web interface, (2) the current system time, and (3) an allowance of 2 h for NCEP to disseminate any model run. To insure WRFEE has the large-scale initialization data to cover a model “spin-up” period, 6 h is subtracted from the intermediate GFS base reference hour yielding the actual GFS base reference hour. WRFEE then fetches the GFS data from NCEP based on the calculated base reference hour and extracts the 0-, 3-, 6-, 9-, 12-, 15-, 18-, 21-, 24-, 27-, and 30-h forecasts, which covers 6 h of model spin-up and 24 h of forecast.

Initial as well as ending year, month, day, and hour values are then determined since these values are required for the WPS and WRF namelist files. These namelist files provide information to the WPS and WRF programs and include such information as start/end forecast times, the size of the domain, the grid resolution, etc.

A configuration file, “runwpsplus.config”, required by “RUNWPSPLUS” (WPS and a Perl script) is generated next. (Note: a detailed look at the RUNWPSPLUS Perl module is beyond the scope of this document. For a closer examination of RUNWPSPLUS, it is recommended that the

user obtain the document, “*RUNWPSPLUS User’s Manual*” from Dr. Brian Reen.* This is a partial list of what information the configuration file sets parameters for:

1. Start and end dates in the determination of what MADIS files to retrieve
2. The grid geometry to use, such as grid spacing, number of points in x-, y-direction, etc.

The execution of the Perl module, “runwpsplus.pl”, consists of these steps:

1. Fetches MADIS observations starting from 3 h prior to forecast start (to enable model spin-up) going out 9 h for a total observation time period of 12 h. The observation types include:
 - a. mesonet—various networks of surface monitoring stations
 - b. maritime—ship reports
 - c. meteorological terminal aviation routine weather report (METAR)—meteorological aerodrome report; surface report typically coming from airports
 - d. profiler—vertical profile of winds and temperature originating from the National Oceanic and Atmospheric Administration (NOAA) Profiler Network (NPN)
 - e. rawinsonde observation (RAOB)—an upper-air observation
 - f. surface aviation observation (SAO)
2. Fetches real-time global sea surface temperatures from the National Weather Service (NWS) Environmental Modeling Center.
3. Can fetch snow cover data generated from the Snow Data Assimilation System (SNODAS) modeling and data assimilation system, available at NOAA NWS National Operational Hydrologic Remote Sensing Center (NOHRSC); however, since the Department of Defense (DOD) blocks File Transfer Protocol (FTP) requests, this is not currently enabled within RUNWPSPLUS. Note that it can be turned on if your facility is able to use FTP. Examine runwpsplus.pl to see how to enable this option.
4. Runs the various aspects of WPS:
 - a. madis_to_little_r.exe: converts the MADIS observations to the so-called Little-R format (Little-R format is a remnant from the earlier generation mesoscale model, “MM5”). The output file is pre_obsgrid_single_littler.txt_prefilter.†
 - b. littler_filter.exe: takes the *prefilter file as input and performs user-defined (via use/reject lists) quality control checks on the observations.
 - c. obsgrid_sort_only.exe: sort the data chronologically; remove duplicates.

*U.S. Army Research Laboratory (ARL) Computational and Information Sciences Directorate (CISD) BED; brian.p.reen.civ@mail.mil.

† MM5 is 5th Generation Mesoscale Model.

- d. obsgrid.exe—obsgrid performs additional quality checks on the observations (5):
 - i. Gross error checks are made (such as verifying all values are within a prescribed range, that pressure decreases with height, etc.).
 - ii. Remove spikes from temperature and wind profiles.
 - iii. Adjust temperature profiles to remove superadiabatic layers.
 - iv. Note that no comparisons are made to other reports or to the first-guess field.
 - v. Also, the analysis is enhanced using observations.
- e. At the completion of runwpsplus.pl, these are the key products:
 - i. OBS_DOMAIN101, OBS_DOMAIN201, OBS_DOMAIN301 files that represent the MADIS observations for the three nested domains, that have passed strict quality control, and are time-ordered and formatted to be read-in by the WRF program. These files are symbolically linked into the WRF “working” directory, (WRF_HOME)/test/em_real/, and will be used during WRF FDDA when the WRF forecasts are dynamically “adjusted” to the observation values in these files.
 - ii. “met_em*” files that represent GFS data that has been horizontally interpolated onto the simulation domains defined in the namelist file for WPS (6). These “met_em” files are then symbolically linked into a WRF working directory and are used to generate WRF initial and boundary conditions.

Once the met_em* files are ready, the WRF “real” program may be run. “real” vertically interpolates the data from “met_em*” files, creates boundary and initial condition files, and performs consistency checks (7).

Last, a WRF FDDA 24-h forecast is run, using WRF compiled in DMPAR mode. (Note: it is very important that if one wishes to use the parallel aspects of WRF with FDDA, then the DMPAR version must be used because WRF FDDA does not execute properly when SMPAR is employed—as of the time of this writing). The version of WRF used with WRFEE, version 3.4.1, has been modified by Dr. Brian Reen* and differs from the standard WRF distributed by NCAR as follows:

1. Engineered a fix for overly dry conditions introduced by observation nudging.
2. Installed a fix for two bugs related to vertical interpolation of the innovation profile in observation nudging code.
3. The surface observation in rawinsondes is detached from the sounding and is now assimilated as the surface observation that it is for observation nudging.

* See footnote on page 5.

4. The Mellor-Yamada-Janjic background turbulent kinetic energy value is decreased from 0.10 to 0.01 J/kg and the planetary boundary layer depth diagnosis methodology output for use in observation nudging (and model output) is modified.

Listed below are parts of a WRF version 3.4.1 namelist file for the 15.75-/5.25-/1.75-km nested domain (see figure 2). This namelist file is read by both the “real” program and the “WRF” program. Note the various blocks that are preceded by a “&” to designate each section. The main blocks are those for (1) “time_control” –to set the forecast time span, etc., (2) “domains” –to outlay the grid geometry, (3) “physics” –to set parameters for various schemes, such as longwave radiation, (4) “fdda” –to set all the parameters used in four-dimensional data assimilation, and (5) “dynamics” –for items such as damping coefficients. Note in the fdda section that obs_nudge_opt is turned on (i.e., set to 1) for each domain and that the “fdda_start” and “fdda_end” values are respectively 0 and 420 (minutes), with obs_dtramp=60. In other words, observation nudging will begin at the start of the forecast period and continue for 6 h, then for the next 60 min, the influence of the observations on the forecast will begin to ramp down. Also note that obs_nudge_wind(temp/mois) are all turned on and that associated with each one of those is an obs_coef_* value, which are factors related to how strongly one intends the observation to influence the forecast fields. For a detailed look at each namelist type, it is recommended that the user go to the WRF homepage: <http://www.mmm.ucar.edu/wrf/users/>.

```

&time_control
  run_days           = 0,
  run_hours          = 00,
  run_minutes        = 0,
  run_seconds        = 0,
  start_year         = 2013, 2013, 2013,2013
  start_month        = 09, 09, 09,09
  start_day          = 26, 26, 26,26
  start_hour         = 00, 00, 00,00
  start_minute       = 00, 00, 00,00,
  start_second       = 00, 00, 00,00,
  end_year           = 2013, 2013, 2013,2013
  end_month          = 09, 09, 09,09
  end_day            = 27, 27, 27,27
  end_hour           = 00, 00, 00,00
  end_minute         = 00, 00, 00,00
  end_second         = 00, 00, 00,00
.
.
.
/

&domains
.
.
.
e_sn                = 113,145,289,
e_we                = 113,145,289,
dx                  = 15750,5250,1750,
dy                  = 15750,5250,1750,
.
.
.
/

&physics
.
.
.
mp_physics          = 8,      8,      8,8
ra_lw_physics       = 1,      1,      1,1
ra_sw_physics       = 1,      1,      1,1
sf_sfclay_physics   = 2,      2,      2,2
sf_surface_physics  = 2,      2,      2,2
sf_urban_physics    = 0,      0,      0,0
bl_pbl_physics      = 2,      2,      2,2
.
.
.
/

&fdda
grid_fdda           =0,0,0,0
obs_nudge_opt       =1,1,1,1,

```

Figure 2. Sample namelist.input file for 15.75-/5.25-/1.75-km nested domain configuration (continues on next page).

```

fdda_start                =0.,0.,0.,0.
fdda_end                  =420.,420.,420.,420.
obs_nudge_wind            =1,1,1,1,
obs_coef_wind             =4.E-4,4.E-4,4.E-4,4.E-4
obs_nudge_temp            =1,1,1,1
obs_coef_temp             =4.E-4,4.E-4,4.E-4,4.E-4
obs_nudge_mois            =1,1,1,1
obs_coef_mois             =4.E-4,4.E-4,4.E-4,4.E-4
obs_dtramp                =60.,
.
.
.
/

&dynamics
.
.
.
w_damping                 = 1,
.
.
.
zdamp                     = 7000.,7000.,7000.,7000.
.
.
.
damp_opt                  = 0,
/

&bdy_control
spec_bdy_width            = 5,
spec_zone                 = 1,
relax_zone                = 4,
spec_exp                  = 0.0,
specified                 = .true., .false.,.false.,
nested                    = .false., .true., .true.,
/

&namelist_quilt
nio_tasks_per_group = 0,
nio_groups = 1,
/
&grib2
/

&dfi_control
dfi_opt = 0,
/

```

Figure 2. Sample namelist.input file for 15.75-/5.25-/1.75-km nested domain configuration (continued).

When the WRF FDDA run is complete, the WRFEE WS will return a string to the Web interface indicating that processing is complete and that the WRF output files are ready. The output product, wrfout(date) files, will be resident in the (WRF_HOME)/test/em_real directory.

Note that as the entire WRFEE process plays out, the Glassfish log in (GLASSFISH-INSTALL-DIR)/glassfish(version #)/glassfish/domains/domain1/logs/server.log enables the user to track the progress.

5. Obtaining the Software

For those interested in obtaining this software, please contact the lead author (575-678-7474). You will be supplied with a WAR file containing the WS code, the WRF Preprocessing System code module (and its associated Perl script, runwpsplus.pl) and the WRF module, as well as build instructions.

6. References

1. MVC for Noobs. <http://net.tutsplus.com/tutorials/other/mvc-for-noobs> (accessed 10/07/13).
2. The Java EE 6 Tutorial. <http://docs.oracle.com/javaee/6/tutorial/doc/bnaeo.html> (accessed 10/7/13).
3. The Library Documentation Generator–Generated Documentation.
<http://docs.oracle.com/javaee/5/javaxserverfaces/1.2/docs/tlddocs/> (accessed 10/7/13).
4. Annotation Type WebMethod.
<http://docs.oracle.com/javaee/5/api/javax/jws/WebMethod.html> (accessed 10/7/13).
5. Chapter 7: Objective Analysis (OBSGRID)
http://www.mmm.ucar.edu/wrf/users/docs/user_guide_V3.1/users_guide_chap7.htm
6. User’s Guide for the NMM Core of the Weather Research and Forecast (WRF) Modeling System Version 3. http://www.dtcenter.org/wrf-nmm/users/docs/user_guide/V3/users_guide_nmm_chap3.pdf (accessed 10/7/13).
7. WRF ARW OnlineTutorial.
<http://www.mmm.ucar.edu/wrf/OnLineTutorial/Basics/WRF/index.html> (access 10/7/13).

List of Symbols, Abbreviations, and Acronyms

ARL	U.S. Army Research Laboratory
BED	Battlefield Environment Division
CISD	Computational and Information Sciences Directorate
DMPAR	distributed memory–parallel
DOD	Department of Defense
FDDA	four-dimensional data assimilation
FTP	File Transfer Protocol
GFS	Global Forecast System
GMT	Greenwich Mean Time
GUI	graphical user interface
HTML	Hypertext Markup Language
JSF	JavaServer Faces
MADIS	Meteorological Assimilation Data Ingest System
METAR	meteorological terminal aviation routine weather report
MPI	Message Passing Interface
MVC	model-view-controller
NCEP	National Centers for Environmental Prediction
NOAA	National Oceanic and Atmospheric Administration
NOHRSC	National Operational Hydrologic Remote Sensing Center
NPN	NOAA Profiler Network
NWS	National Weather Service
OpenMP	open multiprocessor
Perl	Practical Extraction and Report Language
RAOB	rawinsonde observation

SAO	surface aviation observation
SMPAR	shared memory–parallel
SNODAS	Snow Data Assimilation System
UTC	Coordinated Universal Time
WAR	Web archive
WPS	WRF Preprocessing System
WRF	Weather Research and Forecasting
WRFEE	WRF end-to-end
WS	Web service
XHTML	Extended Hypertext Markup Language

<u>No. of Copies</u>	<u>Organization</u>
1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
2 (PDFs)	DIRECTOR US ARMY RSRCH LAB RDRL CIO LL RDRL IMAL HRA RECORDS MGMT
1 (PDF)	GOVT PRINTG OFC A MALHOTRA
5 (PDFs)	DIRECTOR US ARMY RSRCH LAB RDRL CIE M R E DUMAIS B P REEN D I KNAPP RDRL CIE D S G OBRIEN S F KIRBY